Pixel-cluster Decomposition and Tracking for Multiple IR-Sensor Surveillance

Sabino Gadaleta, Aubrey Poore, and Benjamin J. Slocumb

Numerica Corporation, P.O. Box 271246, Fort Collins, CO 80527

ABSTRACT

Tracking midcourse objects in multiple IR-sensor environments is a significant and difficult scientific problem that must be solved to provide a consistent set of tracks to discrimination. For IR sensors, the resolution is limited due to the geometry and distance from the sensors to the targets. Viewed on the focal plane for a single IR sensor, the targets appear to transition from an unresolved phase (merged measurements) involving pixel clusters into a mostly resolved phase through a possibly long partially unresolved phase. What is more, targets can appear in different resolution phases at the same time for different sensors. These resolution problems make multi-sensor tracking most difficult. Considering a centralized multi-sensor tracking architecture we discuss robust methods for identification of merged measurements at the fusion node and develop a method for merged measurements processing and pixel cluster parsing that allows the tracking system to re-process raw merged measurement data for improved tracking performance. The resulting system can avoid inconsistent measurement data at the fusion node. We then present a more general multiple hypothesis pixel-cluster decomposition approach based on finding k-best assignments and solving a number of n-dimensional assignment problems over n frames to find a decomposition among several pixel-cluster decomposition hypotheses that best represents a frame of data based on the information from n frames of data.

Keywords: Infrared Sensor Surveillance, Pixel (Clump) Cluster Tracking, Single and Multi-Assignments, Pixel-Cluster Decomposition, Multiple Hypothesis Pixel-Cluster Decomposition

1. INTRODUCTION

The aim of the Ballistic Missile Defense System (BMDS) is to provide a layered defense against ballistic missile threats. A space-based surveillance system, consisting of multiple satellites with visible and infrared sensors, is envisioned as a major component. A central component is the tracking system which has to provide accurate tracks for discrimination and prediction of the threat's position. Countermeasures in the form of debris, chaff, spent fuel, and balloons produce large numbers of Closely Spaced Objects (CSOs) that need to be tracked accurately to support sensor pointing and discrimination. The tracking system receives measurement data from Signal Processing (SP). For visible/infrared surveillance systems, object information is typically recorded as a digital image in a solid state array detector such as a Charge-Coupled Device (CCD). The task of SP *digital image restoration*[1] algorithms is to recover information that was lost during the image acquisition process for example due to optical blurring. After performing image restoration methods edge-connected components, or *pixel-clusters*, can be identified in the image. Given the presence of CSOs and the influence of object resolution limiting factors, the signal from multiple objects can overlap to produce a pixel-cluster that represents multiple objects. One can distinguish between Resolved Object (RO) pixel-clusters, n-CSO pixel-clusters, and Unresolved Closely-Spaced Object (UCSO) pixel-clusters, where a RO pixel-cluster represents the signal of a single point source, a n-CSO pixel-cluster represents the signal from n point sources, and an UCSO pixel-cluster represents the signal from an unknown number of objects. An UCSO pixel-cluster is also often referred to as a *clump*.

In the presence of UCSOs tracking performance can suffer greatly since multiple sensor types, geometric views, signal-to-noise ratio, and other factors imply that the same set of targets may have different levels of object appearance at the same time from the same sensor or from multiple platforms. The frame-to-frame measurement inconsistency on a single-sensor and the measurement inconsistency between frames from different platforms produces many challenges for

Further author information: (Send correspondence to S.G. or A.P.)
S.G.: E-mail: smgadaleta@numerica.us, Telephone: (970) 419 8343
A.P.: E-mail: abpoore@numerica.us, Telephone: (970) 419 8343

a tracking system. A group of targets might appear resolved on the focal plane of one sensor and completely unresolved on the focal plane of another sensor. This can severely degrade the accuracy of the global track picture, produce mono-view tracks at the fusion node, and cause measurement misassociations that negatively impact discrimination performance. For a single-sensor, an inconsistent resolution phase, where pixel-clusters break up an then re-clump, may persist over long periods and cause the tracking system to initiate tracks on subclusters that starve and become inaccurate or are dropped once the objects re-clump. The result may be redundant and spurious track initiations and contention for measurements.

This paper reports on specific *pixel-cluster decomposition and tracking methods* that are designed to improve tracking performance in the presence of UCSOs. The methods include the detection of candidate UCSO measurements, multi-assignment of UCSOs to tracks, and decomposition of UCSO pixel-clusters into subclusters consistent with the set of global track states. These methods are designed to *provide accurate tracks on RO and UCSO pixel-clusters as observed by the most resolving sensor in the system for a particular group of CSOs*. The methods reduce measurement contention and can prevent the corruption of tracks on resolved objects through unresolved measurements. In special cases, where only a single sensor can fully resolve a group of targets, the pixel-cluster tracking methods allow us to obtain accurate multi-view tracks on all the resolved objects if a second sensor reports the measurements as unresolved signals. Thus, the pixel-cluster tracking methods can greatly reduce the requirement for high-resolution sensors in the system.

In many cases, none of the sensors will be able to fully resolve a group of CSOs throughout most of the scenario. One frequently observes that objects remain unresolved through most of midcourse only separating close to re-entry. To support early discrimination, advanced signal processing methods are required that can super-resolve the unresolved image data to produce resolved object returns as quickly as possible. Super-Resolution (SR) algorithms refer to SP techniques that attempt to produce a restored image in which details too fine to be resolved by the optical system are revealed, or super-resolved.[2] One can distinguish super-resolution methods that operate with information from a single low-resolution image (*SR image restoration*) or with information from multiple low-resolution images (*SR image reconstruction* and *image fusion*) that represent different "looks" at the same scene.[3] For the tracking of point targets, this problem is related to the so-called *CSO problem* that addresses the super-resolution of point targets.[4–6] The presented pixel-cluster decomposition algorithm can be extended to this purpose if the optical Point-Spread Function (PSF), that describes how a single point signal is observed in the optical system, is known and future work will address the cooperation between tracking and SP for SR image reconstruction.

This paper is structures as follows: Section 2 gives a brief introduction to relevant terms from digital image processing; Section 3 discusses some of the tracking challenges in the presence of UCSOs; Section 4 discusses the pixel-cluster tracking methods developed to improve tracking in the presence of UCSOs; Section 5 suggests an extension of the pixel-cluster decomposition approach that considers multiple pixel-cluster decompositions for a frame and that finds the best decomposition based on the information over multiple frames. This approach is similar in concept to multiple frame cluster tracking that associates between multiple clustering hypotheses between frames as discussed in other work.[7, 8] Section 6 concludes the paper.

2. DIGITAL IMAGE PROCESSING FOR SPACE-BASED OPTICAL SURVEILLANCE

Space-based or air-based visible and infrared sensors are an important component of the BMDS since they provide a global end-to-end coverage of ballistic missiles launches. Such sensors typically receive object signals as digital images in the sensor's focal plane. A digital image is composed of a finite number of elements, or *pixels*, that transform incoming energy into a voltage waveform. The typical arrangement found in digital cameras is a *focal plane array* that packages several thousand pixels into a square grid (see Figure 1). The energy emitted by the source objects is referred to as the intensity J in units of watts.

The amount of energy incident on the sensor per unit area is referred to as the *irradiance* I and is proportional to intensity J, $I \sim J/d^2$, where d is the distance between the source and the sensor. The *sensitivity* of the sensor gives the minimum irradiance that can be detected. The pixel image is a function of the sum over the irradiance incident on the pixel's sensing material. A pixel with incident irradiance above the sensitivity threshold represents a *signal pixel*. The smallest signal that can be measured is a signal whose level is equal to the noise accompanying that signal, i.e., a Signal-to-Noise Ratio (SNR) of one. In CCD imaging, SNR refers to the relative magnitude of a signal compared to the uncertainty in that signal on a per-pixel basis. An important component is the shot noise that originates from the signal itself due to the inherent natural variation of the incident photon flux, since it gives a limit on the smallest size of a pixel:

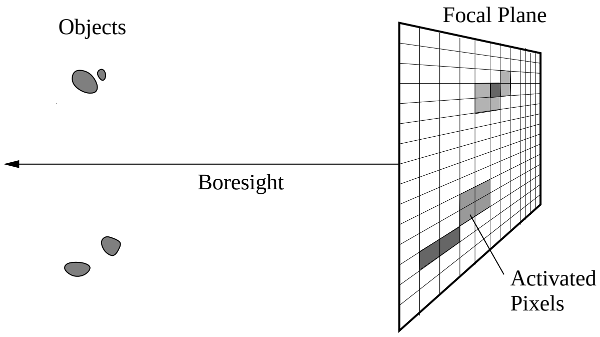


Figure 1: Focal-plane pixel array that observes five objects as three pixel-clusters.

SNR decreases roughly as the square root of the pixel size since the RMS shot noise decreases as the square root of the signal.[9] Unfortunately, a larger pixel size reduces the spatial image resolution. Thus, the optimal pixel size must be chosen from image quality measures.

To improve the quality of a recorded image a variety of image enhancement techniques such as image differencing and filtering techniques can be applied.[10] The image enhancement is designed to remove those pixels from the set of signal pixels that are likely caused by noise, false targets, clutter, etc. The resulting image is composed of a set of *object pixels* that is a subset of the signal pixels. A connected set of object pixels is referred to as a *pixel-cluster*, or *clump*. The segmentation of the object pixels into pixel-clusters is performed through *image segmentation* algorithms. Figure 1 shows a set of objects that are observed in a focal-plane pixel array. In the example, the objects are observed as three pixel-clusters.

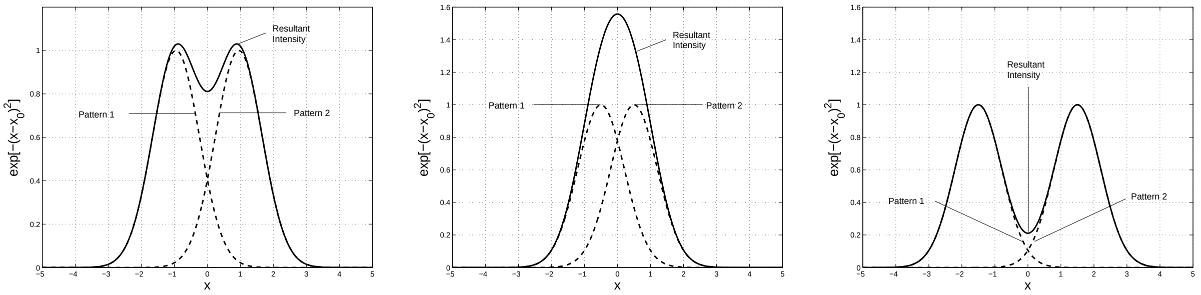
The final step of signal processing in support of tracking is to compute *representative measurement data* that describes the measured object signals in compact form and estimates actual object location and the uncertainty in this estimated location. Given a pixel-cluster, the representative measurement can consist of the centroid position and an intensity weighted covariance.

An important characteristic of any optical system is its *Point Spread Function (PSF)*. The PSF is the output of the imaging system for an input point source.[11] The point source technically represents a single point source at infinity and the PSF describes the *blurring* of the point signal with the amount of blurring being determined for example by the quality of the optical components, atmospheric blur, and motion blur.[10] The PSF of an optical sensor can be computed off-line through modeling and simulation, or on-line through the observation of point targets in the scene such as stars or test launched missiles. Although a PSF will typically vary from pixel to pixel and over time, a two-dimensional Gaussian distribution is often used to approximate the PSF of an optical system.[5] Assuming that x_k, y_k, represent continuous coordinates of a unit amplitude point-source viewed on the focal plane, the PSF of this point source is given by

$$i(x,y) = \frac{1}{2\pi\sigma^2_{psf}} e^{-\frac{(x-x_k)^2-(y-y_k)^2}{2\sigma^2_{psf}}},$$

with *blur width* parameter σ_{psf}.[12]

The original point signal degradations such as optics blurring can be modeled as being the result of *convolution*.[10] Similarly, the process of *image restoration*, e.g., deblurring, can be referred to as *deconvolution*. The process of deconvolution is designed to remove artifacts in the image due to, e.g., the PSF and to improve the resolution of the image. The *spatial resolving power* of an optical instrument is defined as the smallest distance between two object points at which they still may be imaged as separate points,[13] i.e., at which the intensity distributions can be separated by the optical instrument. An often used criterion that specifies when two overlapping patterns are distinguishable is the *Rayleigh criterion*: two object points are considered to be resolved when the central diffraction maximum of the first object coincides with the first diffraction minimum of the second object. Diffraction describes the change of propagation direction of a wave striking an obstacle. An objective lens with a circular aperture provides an image of a single point object as a circular diffraction pattern in the focal plane of the objective. The energy contained within the first ring is about 84% of the total energy. Assuming a Gaussian PSF with width σ_{psf}, the radius of a circle that contains 84% of the total energy is given by 1 R (Rayleigh) := ρ_{min} = 1.9 σ_{psf}. Figure 2 illustrates this resolution criterion. Shown are the intensity of two point sources with $\sigma_{psf} = 1$ together with the total intensity function at a distance ρ between the sources of (a) $\rho = 1.9$, (b) $\rho = 1$, and (c) $\rho = 3$.



(a) (b) (c)

Figure 2. Illustration of the Rayleigh criterion of two point sources separated by a distance ρ. (a) $\rho = 1$R, (b) $\rho < 1$R, and (c) $\rho > 1$R.

The Rayleigh resolution criteria can be related to the notion of super-resolution: an algorithm that is capable of resolving point sources separated less than 1 R is said to have *super-resolution (SR)* capability. Super-resolution can be achieved from a single image through special *SR image restoration*[4–6] methods or from multiple images through *SR image reconstruction*[14] or *SR image fusion*[15] techniques.

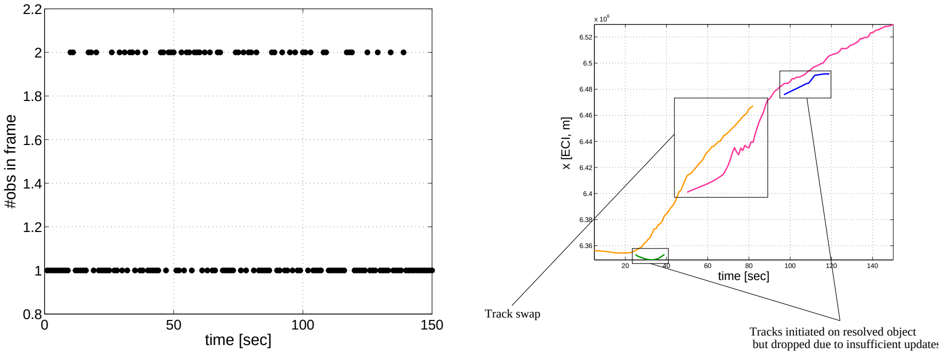
3. TRACKING IN THE PRESENCE OF UNRESOLVED CLOSELY SPACED OBJECTS

When observing Closely Spaced Objects (CSOs), the signal from multiple objects may overlap to produce an Unresolved Closely Spaced Object (UCSO) pixel-cluster, i.e., a merged measurement, even after image restoration techniques have been applied. Although super-resolution methods are available that attempt to decompose an unresolved measurement into resolved object measurements, these algorithms are computationally expensive and often only applicable if a small number of target signals overlap.[16] Thus it can be assumed that UCSO pixel-clusters or clumps will be reported in current surveillance systems as input to tracking. When tracking in the presence of UCSO pixel-clusters some unique challenges have been observed that include an *inconsistent single-sensor* and an *inconsistent multi-sensor object resolution*.

3.1. Inconsistent Single-Sensor Object Resolution

A major problem with the detection of CSOs and low-visible targets on a single-sensor is an *inconsistent frame-to-frame evolution of the resolution process*. On consecutive frames, due to changes in visibility conditions or relative sensor positions, target groups might appear partially resolved, unresolved, or resolved. Threats may transition from an unresolved phase into a resolved phase through a possibly long partially resolved phase. Targets close to the detection threshold level might be detected or missed between frames. To illustrate this issue we simulated three near ballistic targets viewed by a single sensor. From frame-to-frame the sensor either produces an unresolved measurement for all three objects or a resolved measurement for one of the targets and an unresolved measurement for the other two. Figure 3(a) shows the number of observations in a frame as a function of time. We can see that this number frequently varies between 1 and 2. This type of measurement data can produce suboptimal tracking performance if regular one-to-one assignment problem is used. Figure 3(b) shows the x-coordinate of the estimated tracks using a Multiple Frame Assignment (MFA) tracking system with regular one-to-one assignment tracking and a relatively short out-date time[*] of 5 s. In this simulation the tracker initiated three tracks that were eventually dropped again due to lack of updates, thus we observe sub-optimal tracking performance due to an inconsistent single-sensor frame-to-frame object resolution.

[*] The tracker will drop a track if it has not been updated in out-date time seconds.



(a) (b)

Figure 3: (a) Number of observations in a frame as a function of time. (b) x-coordinate of the estimated track states.

3.2. Inconsistent Multi-Sensor Object Resolution

A multi-view tracker (or global tracker, fusion node tracker) receives multiple views at groups of targets. Due to different geometric views at the targets, different distances to the targets, and differences in the physical resolution capabilities of the sensors (e.g., number of pixels in the focal plane array, instantaneous field-of-view of a pixel, etc.), the object resolution from one sensor may be inconsistent with the object resolution from another sensor. What may be observed as resolved objects from one view may remain unresolved from another view. As a result, the multi-view tracker faces two unique challenges: *track starvation* and *track corruption*.

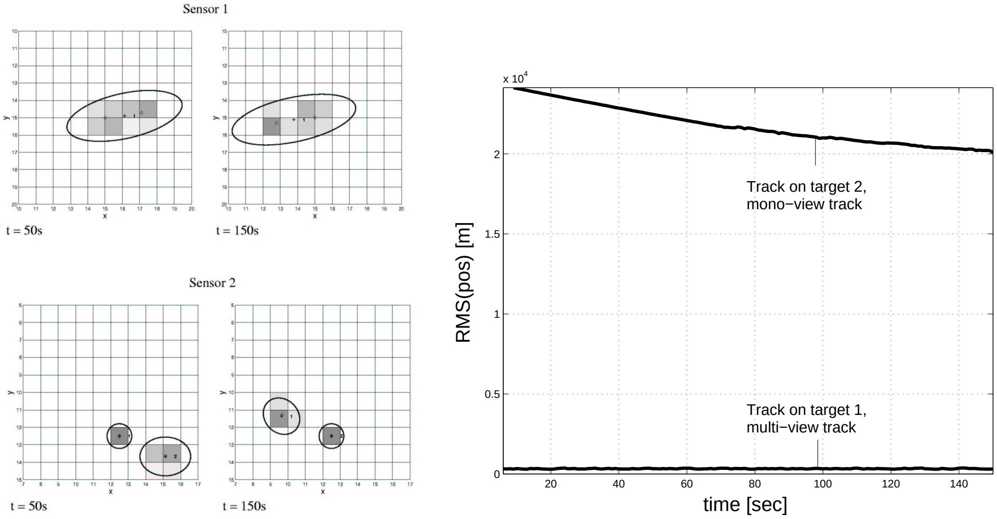
3.2.1. Track Starvation

If only a single sensor can resolve a group of near targets then so-called *mono-view tracks* may persist on a multi-view tracker. The mono-view tracks are typically updated by a single sensor and inaccurate along the unobservable, i.e., range direction. *Multi-view* tracks on the other hand are updated by multiple diverse looks and are typically accurate in range. We illustrate this through a simple example that simulates two near targets viewed by two sensors. Sensor 1 can not resolve the targets and reports a single measurement representing the unresolved objects (no specialized CSO resolving algorithm was used). Sensor 2 one the other hand, due to its different geometric view at the targets, can resolve the two targets. Figure 4(a) shows the focal planes of the two sensors at times $t = 50s$, and $t = 150s$. Figure 4(b) shows the Root Mean Squared (RMS) position error for the two targets based on 10 Monte Carlo runs using an MFA tracking system based on regular one-to-one assignment tracking. We can observe that the multi-view track is far more accurate than the mono-view track.

The goal in avoiding track starvation is to develop methods that prevent the existence of mono-view tracks on a multi-view tracking platform. Clearly this is only possible if at least two sensors look at a common group of targets. Multi-assignment methods[17] have been suggested to improve tracking in this situation. In the following section we will illustrate the performance gain obtained through multi-assignment of UCSO measurements to tracks and discuss a pixel-cluster decomposition algorithm that can improve performance over the results obtained through multi-assignment.

3.2.2. Track Corruption

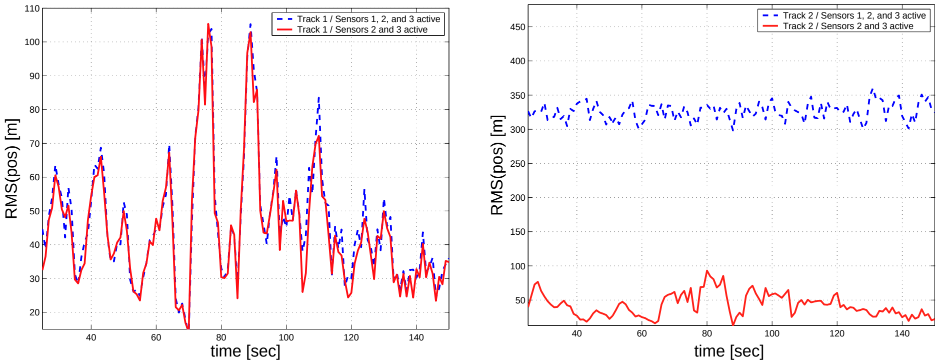
Track corruption is different from track starvation and represents the problem where accurate multi-view tracks on resolved objects are diluted through updates from merged measurements received from an unresolving sensor. The problem is that the unresolved measurements represent multiple objects while the resolved measurements represent a single object. If multiple sensors can resolve a group of objects while some sensors can't, then the update of the multi-view track with an unresolved measurement can corrupt the accuracy of the multi-view track since an update of the track perturbs the track towards the location of other objects. Tracks on resolved objects are of great use for target intercept and discrimination. Thus, the prevention of the corruption of tracks on resolved objects can be very important. The following scenario illustrates the problem. The scenario is an extension of the scenario discussed in the last paragraph with one added sensor



(a) (b)

Figure 4. Scenario to illustrate track starvation problem. (a) Sensor 1 can not resolve the targets, sensor 2 can resolve the targets. (b) RMS position error on the multi-view track and the mono-view track.

that can resolve the objects. Thus the scenario now considers two resolving sensors and one unresolving sensor. In this study we compared the multi-view performance (using regular one-to-one assignment tracking) based on 10 Monte Carlo runs first when input from all three sensors is allowed, and second, when only input from the resolving sensors 2 and 3 is allowed. Figure 5 shows the RMS position error for the two studies for (a) target 1 and (b) target 2. For target 2 we can clearly see that the track is more accurate *without* the input from sensor 1. For this study we found that on target 1, the total RMS position error was 93.5 m with all three sensors active, while it was 58.9 m with only sensors 2 and 3 active. For target 2, the total RMS error was 326 m with all three sensors active, while it was only 51.5 m when only updates from the resolving sensors where allowed.



(a) (b)

Figure 5. Illustration of track corruption. (a) RMS position error on (a) target 1 and (b) target 2 with updates from all three sensors and updates only from resolving sensors.

While track corruption can degrade track accuracy, it also impacts discrimination since the updating of tracks on resolved objects with measurements from unresolved objects dilutes the purity of the observation history that is important for discrimination. As we will show in the following section, multi-assignment methods that can avoid the problem of track starvation actually make the problem of track corruption worse. The best solution to avoid both track starvation and track corruption is a pixel-cluster decomposition of UCSO pixel-clusters to produce measurement sets consistent with the global track database.

3.3. Simulation Environment

The targets in the studies in this paper simulate ballistic missiles separated by 500 m to 1000 m. The sensors simulate infrared sensors with data observed in a focal-plane pixel array. We simulated 30×30 sized pixel-arrays with an instantaneous field-of-view of 5e-5 rad for a single pixel. Including effects of jitter the point-spread function had a blur width of approximately 0.6 pixels. No clutter and no sensor noise was simulated. No image restoration techniques were used to restore the original image. We note that UCSO pixel-clusters are observed even when using image restoration techniques such as the Lucy-Richardson algorithm[18, 19] when targets are closely spaced. The tracking system represents a multiple-frame Assignment tracking system were a gravity compensated nearly-constant velocity filter was used with a process noise level of $\sigma = 10$.

4. PIXEL-CLUSTER TRACKING

To address the challenges presented in Section 3 we present in this section *pixel-cluster tracking algorithms* that can improve tracking performance in the presence of UCSO pixel-clusters. The goal of these pixel-cluster tracking methods is 1) to *obtain a set of measurements throughout the constellation of sensors that is consistent with the global track states*, and 2) to *robustly break large pixel-cluster UCSOs into subclusters to improve track accuracy*.

4.1. Identification of Candidate UCSO Measurements Through Multi-assignment

To minimize the impact of limited sensor resolution on tracking we attempt to compute a set of measurements consistent across the sensors to obtain tracks that are updated by measurements from all sensors that measure a signal from an object a track represents. To this end we need to identify the number of tracked objects a measurement represents. To identify *UCSO candidate measurements*, i.e., measurements that associate with multiple multi-view tracks, we solve an multi-assignment problem of associating measurements to tracks.

An iterative procedure with successive 2-D assignments of decreasing size to solve a multi-assignment problem was developed by Kirubarajan *et al.*[17] Here we apply this scheme to detect UCSO candidate measurements and to identify the number of tracks that the UCSO measurement represents. We assume that a set $\mathcal{T} := \mathcal{T}(t_{k-1}) = \{T_1, \ldots, T_m\}$ of m tracks T_i is established by the tracking system at t_{k-1}. A set $\mathcal{M} := \mathcal{M}(t_k) = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ of n observations $\mathbf{z}_j$ is received at time t_k. We can multi-assign the measurement set $\mathcal{M}(t_k)$ to the track set $\mathcal{T}(t_{k-1})$ through an iterated multi-assignment scheme.[17] In the first round a regular one-to-one assignment between $\mathcal{M}$ and $\mathcal{T}$ is performed. This one-to-one assignment problem is described through the standard two dimensional assignment problem most often used to track objects individually.[7] Now let $\mathcal{U}(\mathbf{z}_j) \subseteq \mathcal{T}$ denote the set of tracks that measurement $\mathbf{z}_j$ assigned to in the current and previous rounds of assignments. After the first round of assignments, the set $\mathcal{U}(\mathbf{z}_j)$ will contain either exactly one or no track. Further let $\mathcal{T}^1 \subseteq \mathcal{T}$ denote the set of tracks that were *not* assigned to a measurement during the first round of assignments. If reliable CSO typing is available we can use this information to adjust the measurement set $\mathcal{M}$ before the second and further rounds of assignments by removing all measurements that represent resolved objects, i.e., that are typed as RO pixel-clusters. Otherwise, the set $\mathcal{M}$ for all rounds of assignments will be left unchanged. After completing the first round of assignments, we wish to find the best one-to-one assignments between the set $\mathcal{T}^1$ and the measurements $\mathcal{M}$. After completing the second round of one-to-one assignments one obtains a track set $\mathcal{T}^2 \subseteq \mathcal{T}^1$ and one can update the sets $\mathcal{U}(\mathbf{z}_j)$ for all measurements $\mathbf{z}_j$ that were assigned to a track during this round. One can continue with multiple rounds of one-to-one assignments until one of the following two conditions is satisfied: (1) $\mathcal{T}^i = \emptyset$, i.e., measurements were assigned to all established tracks, or (2) $\mathcal{T}^i = \mathcal{T}^{i-1}$, i.e., the last round of assignments did not produce new measurement-to-track associations.

The result of the rounds of one-to-one assignments between established tracks and measurements are sets of tracks $\mathcal{U}(\mathbf{z}_j) = \{T_{j1}, \ldots, T_{jn(j)}\}$ that identify the tracks that a measurement is (multi-) assigned to. Thus, the number of objects under track, $n_{\text{obj}}(\mathbf{z}_j)$, that a measurement $\mathbf{z}_j$ represents (assigns to) is given by

$$n_{\text{obj}}(\mathbf{z}_j) = |\mathcal{U}(\mathbf{z}_j)|,$$

where $|\mathcal{U}|$ indicates the number of elements in the set $\mathcal{U}$. This information allows us to identify *candidate UCSO measurements* as measurements with $n_{\text{obj}}(\mathbf{z}) > 1$. This does not imply that measurements that assign to a single track are likely to represent resolved objects. Resolved object typing can only be performed by sensor signal processing or discrimination including expected measurement information for resolved objects. However, if a measurement assigns to multiple tracks, this measurement is likely to represent multiple closely spaced objects. The pixel-cluster tracking methods discussed in the remainder of this section perform special processing of candidate UCSO measurements to improve tracking performance in the presence of closely spaced objects and UCSO pixel-clusters.

4.2. Multi-Assignment of Candidate UCSOs to Tracks

This method is appropriate if no image data in the form of pixel-cluster single pixel irradiances is available with a measurement as may be the case due to communication limitations. Using multi-assignment we can improve tracking performance in the presence of UCSOs by letting a candidate UCSO measurement $\mathbf{z}_{\text{UCSO}}$ update all tracks that assigned to this measurement. This set of tracks is given by the set $\mathcal{U}(\mathbf{z}_{\text{UCSO}})$ that is computed during the multi-assignment procedure as discussed in Section 4.1. We demonstrate simulation results for the performance of this multi-assignment procedure in Section 4.4.

4.3. UCSO Pixel-Cluster Parsing

If pixel-cluster image data is available with a candidate UCSO measurement $\mathbf{z}_{\text{UCSO}}$ we can decompose the pixel-cluster into $n_{\text{obj}}(\mathbf{z}_{\text{UCSO}})$ returns with the goal to produce measurements consistent with the tracked objects represented by the tracks in the set $\mathcal{U}(\mathbf{z}_{\text{UCSO}})$. This *pixel-cluster parsing* allows us to improve tracking performance in the presence of UCSOs compared to the performance achievable by multi-assignment of candidate UCSO measurements to tracks as discussed in the previous paragraph. The pixel-cluster decomposition algorithm we developed for this problem is performed in two steps: (1) distribute particles over the pixel-cluster according to a distribution that reflects the pixel irradiances, and (2) cluster the set of particles into n_{obj} centroids and covariances. The clustering will produce n_{obj} representative measurements on the focal plane of the sensor that represent the pixel-cluster through n_{obj} centroids and covariances that measure the extent of the n_{obj} subclusters. Figure 6 illustrates this algorithm for a sample pixel-cluster that consists of 21 signal pixels as illustrated in Figure 6(a). In this example three tracks assigned with this pixel-cluster, i.e., $n_{\text{obj}} = 3$.

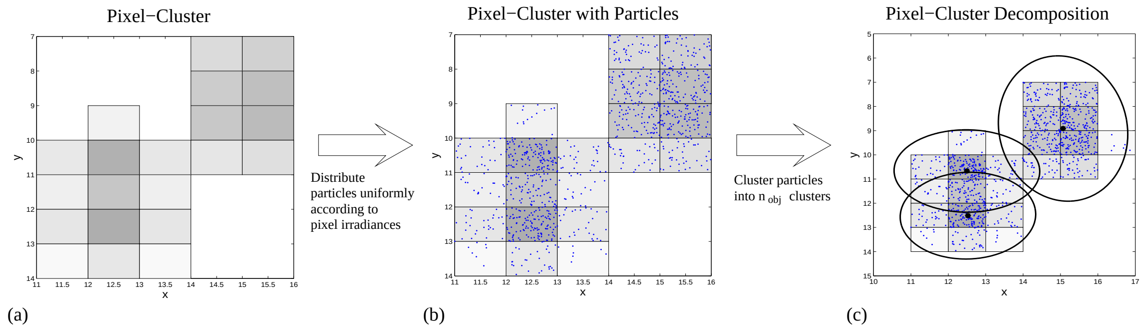

Pixel−Cluster
Pixel−Cluster with Particles
Pixel−Cluster Decomposition
Distribute particles uniformly according to pixel irradiances
Cluster particles into n obj clusters
x
y
(a)
(b)
(c)

Figure 6. (a) Pixel-cluster that consists of 21 signal pixels. (b) 1000 particles distributed over the pixel-cluster to match the irradiance distribution of the pixel-cluster. (c) Clustering of particles and resulting pixel-cluster decomposition into 3 clusters. The covariances are shown at 3 σ level.

For convenience we represent the pixel-cluster through the pixel irradiances by a square matrix I_{ij} which contains zeros for elements ij that are not part of the pixel-cluster. Next we compute the number of particles to distribute over each pixel. We denote with $m_{\text{total}}^{\text{part}}$ the total number of test particles we wish to distribute over the pixel-cluster. Further let $I_{\text{total}}(\mathbf{z}_{\text{UCSO}}) = \sum_{ij} I_{ij}(\mathbf{z}_{\text{UCSO}})$ denote the sum over all single-pixel irradiances of the pixel-cluster corresponding to measurement $\mathbf{z}_{\text{UCSO}}$. Then, the number of particles m_{ij} to distribute over pixel ij is given by

$$m_{ij} = \text{ceil}\left(m_{\text{total}}^{\text{part}} \frac{I_{ij}}{I_{\text{total}}} \right),$$

where ceil denotes rounding to the next largest integer. We distribute the m_{ij} particles uniformly over the pixel ij. Figure 6(b) shows 1000 particles distributed over the pixel-cluster. In the final step we partition the set of particles into n_{obj} clusters through the use of a clustering algorithm. This will decompose the original pixel-cluster into n_{obj} subclusters.

Although any clustering algorithm can be used we restrict the presentation here to the use of the Expectation-Maximization (EM) algorithm.[20] Using the EM algorithm, the above scheme finds a solution that maximizes the likelihood that the pixel-cluster signal represents an n_{obj} mixture and solves for the corresponding n_{obj} distributions. Let $p(\mathbf{z}_{\text{UCSO}})$ denote the probability density function that describes the distribution of the observed pixel-cluster irradiance signal. The particle set $\mathcal{P}(\mathbf{z}_{\text{UCSO}}) := \{\mathbf{x}_1, \ldots, \mathbf{x}_{m_{\text{total}}^{\text{part}}}\}$ represents a sample of data that is drawn from the distribution $p(\mathbf{z}_{\text{UCSO}})$. The EM algorithm allows us to find a model $\boldsymbol{\theta}^*$ that maximizes the log-likelihood given the data

$$L(\boldsymbol{\theta}|\mathbf{z}_{\text{UCSO}}) \approx L(\boldsymbol{\theta}|\mathcal{P}(\mathbf{z}_{\text{UCSO}})) = \sum_{k=1}^{m_{\text{total}}^{\text{part}}} \ln\left(p(\mathbf{x}_k|j_k;\boldsymbol{\theta})P_{j_k}\right),$$

given a mixture of J components, $j_k \in \{1, 2, \ldots, J\}$, and P_{j_k} denoting the prior probability of mixture component j_k. For a Gaussian mixture it is

$$p(\mathbf{x}_k|j;\theta) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{\|\mathbf{x}_k - \boldsymbol{\mu}_j|^2}{2\sigma_j^2}\right),$$

with mean $\boldsymbol{\mu}_j$ and variance σ_j. For pixel-cluster decomposition, the number of components is given by n_{obj}, i.e., $J = n_{\text{obj}}$.

Figure 6(c) shows the pixel-cluster decomposition of the pixel-cluster shown in Figure 6(a) into $n_{\text{obj}} = 3$ clusters that results from a clustering of the particles shown in Figure 6(b) using the EM algorithm. The following section illustrates the impact on tracking performance in the presence of UCSOs when using multi-assignment and pixel-cluster parsing.

4.4. Solving the CSO Resolution Problems

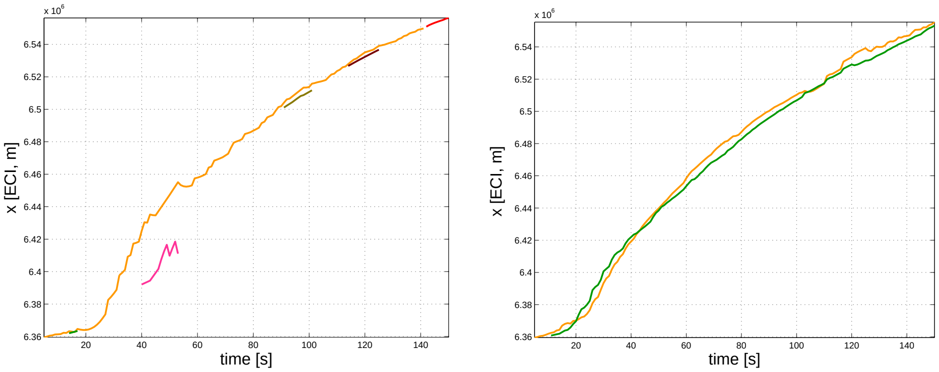
Section 3 illustrated a number of challenges produced by the presence of UCSO pixel-clusters. In this section we demonstrate the improved tracking performance achieved through multi-assignment and pixel-cluster decomposition.

4.4.1. Inconsistent Single-Sensor Resolution

In Section 3.1 we demonstrated the potential impact on tracking performance in the presence of an inconsistent single-sensor frame-to-frame resolution for a scenario with three closely spaced objects. While two of the objects were never resolved, one of the objects was resolved on some frames. When tracking in the presence of inconsistent frame-to-frame measurements we did observe frequent reinitation of a track when the resolved measurement was received over a number of frames and dropping of a track when the resolved object becomes unresolved again and is not resolved within the out-date time of the tracker as shown in Figure 7(a) for a particular Monte-Carlo run. One solution to this problem is to use a longer out-date time in the tracking system, extrapolating tracks over long periods of time in the hope that they may receive a measurement. Long out-date times can produce suboptimal performance since tracks that are not observed any longer or tracks that were falsely initiated are propagated over long periods of time. The propagation can inflate the false track's covariance to a point where it will steal measurements from other tracks. Based on 10 Monte Carlo runs when using an out-date time of 5 s we observed an average of 5.9 tracks per run, out-dating after 10 s we observed 2.7 tracks, out-dating after 15 s we observed 2.2 tracks, and only after increasing the out-date time to 25 s did we observe consistently 2 tracks for all Monte Carlo runs. A better solution than increasing the out-date time is to multi-assign candidate UCSO measurements to tracks or to decompose candidate UCSO pixel-clusters. Figure 7(b) shows the x-coordinate of the estimated tracks when performing pixel-cluster decomposition of detected candidate UCSO measurements. Both multi-assignment and pixel-cluster decomposition allowed us to produce consistently two tracks for all 10 Monte Carlo runs even at a 5 s out-date time.

4.4.2. Inconsistent Multi-Sensor Resolution and Track Starvation

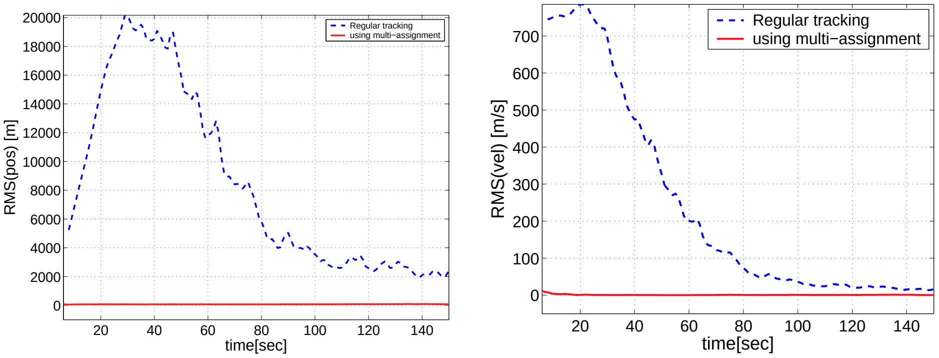
We illustrated the potential for track starvation in the presence of inconsistent multi-sensor resolution of CSOs in Section 3.2. Here we illustrate the improvement gained through multi-assignment and pixel-cluster break-up. All results are based on 10 Monte-Carlo runs.



(a) (b)

Figure 7. (a) Suboptimal tracking performance caused by inconsistent frame-to-frame resolution. (b) Improved tracking performance obtained through performing candidate UCSO pixel-cluster decomposition.

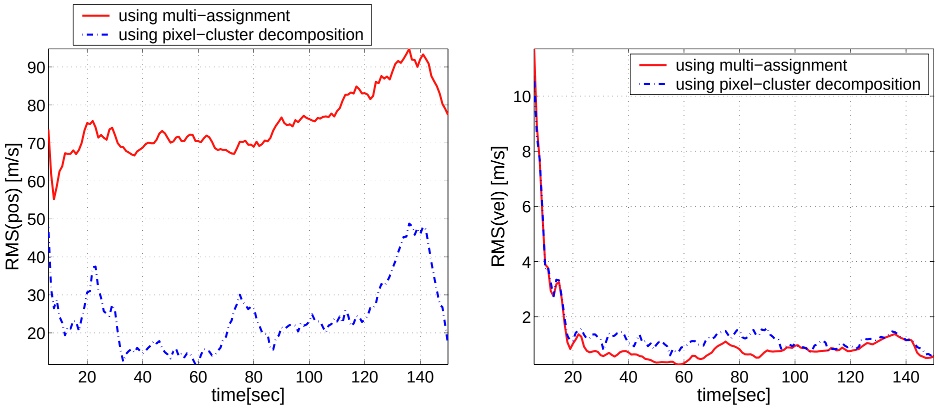
Study 1. We first compare performance on a simple scenario that considers two closely spaced objects as discussed in Section 3.2. The scenario simulates two sensors with different geometric views at the two CSOs. While one sensor can fully resolve the two objects during the course of the scenario the second sensor can't resolve the two objects during the entire flight time. Regular one-to-one assignment tracking produces one accurate multi-view track and one inaccurate mono-view track as illustrated in Section 3.2. Multi-assignment and pixel-cluster decomposition can greatly improve the accuracy of the mono-view track. Figure 8 compares the RMS estimation accuracy on the mono-view track for (a) position, and (b) velocity, obtained with regular tracking and with pixel-cluster tracking using multi-assignment of detected candidate UCSOs to tracks. We did not include any object typing information in the multi-assignment procedure. We can observe that the multi-assignment of the candidate UCSO measurement to the mono-view track gives a dramatic improvement in both position and velocity estimation accuracy.



(a) (b)

Figure 8. Comparison of RMS estimation accuracy on the mono-view track for (a) position, and (b) velocity, obtained with regular tracking and with pixel-cluster tracking using multi-assignment of detected candidate UCSOs to tracks.

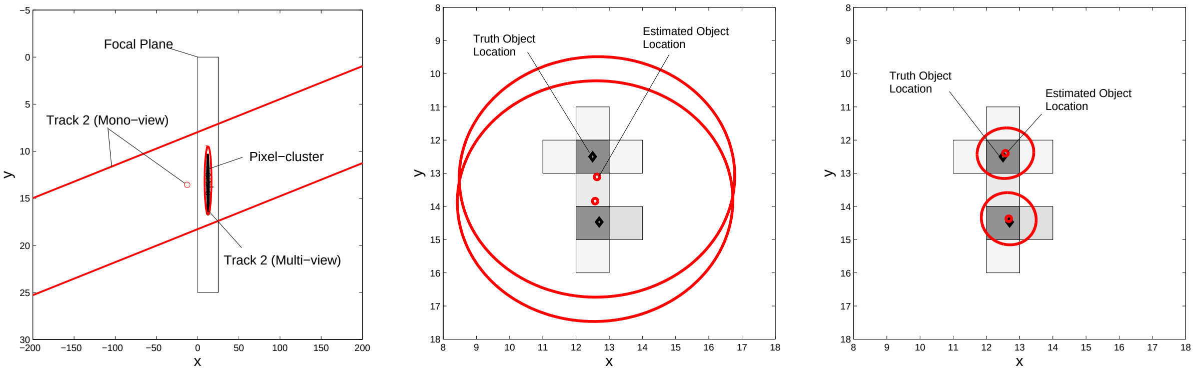
The estimation accuracy can further be improved by performing pixel-cluster decomposition as discussed in Section 4.3. Figure 9 compares the (a) position, and (b) velocity, estimation accuracy on the mono-view track. The pixel-cluster decomposition allows us to considerably improve the position estimation accuracy when compared to the accuracy using multi-assignment. In this particular example (using 100 particles), the accuracy in velocity, using pixel-cluster decomposition, is slightly less than using multi-assignment. In the example in Study 2 below, both position and velocity estimation accuracy are improved with pixel-cluster decomposition and we will see that the decomposition performance improves with the number of particles used for pixel-cluster decomposition.



(a) (b)

Figure 9: Comparison of (a) position, and (b) velocity, estimation accuracy on the mono-view track.

In Figure 10 we compare the estimated track states transformed into the focal-plane of the unresolving sensor at $t = 150$ s for (a) regular tracking, (b) pixel-cluster tracking with multi-assignment, and (c) pixel-cluster tracking with pixel-cluster decomposition. All covariances are shown at 3-sigma level. As we can see in Figure 10(a), the track state estimate on the mono-view track are so inaccurate that the state estimate does not project into the focal-plane of the sensor. Thus, in the presence of UCSOs and unresolving sensors, it will be difficult to associate global track states to the pixel-clusters on the unresolving sensors if no multi-assignment or pixel-cluster decomposition is performed. Figure 10(b) shows that the multi-assignment gives two accurate tracks maintained close to the reported centroid of the pixel-cluster. With pixel-cluster parsing (see Figure 10(c)) we obtain two accurate tracks maintained of the subcluster components of the pixel-cluster.



(a) (b) (c)

Figure 10. Estimated track states transformed into the focal-plane of the unresolving sensor at $t = 150$ s for (a) regular tracking, (b) pixel-cluster tracking with multi-assignment, and (c) pixel-cluster tracking with pixel-cluster decomposition.

Study 2. This second study considers a scenario similar to the one discussed in Study 1 but with four CSOs that are fully resolved by Sensor 1 but remain unresolved during the scenario by Sensor 2. Based on 10 Monte-Carlo runs we compared the position estimation accuracy, the velocity estimation accuracy and the average run-time. Table 1 shows these values for one-to-one assignment tracking, pixel-cluster tracking using multi-assignment and pixel-cluster tracking using pixel-cluster decomposition with the EM algorithm. We also compare the performance of the EM based pixel-cluster decomposition when using different numbers of particles to distribute over the pixel-cluster as discussed in Section 4.3. We used 10 iterations for convergence of the EM clustering algorithm. We can observe that the pixel-cluster decomposition outperforms the multi-assignment performance in all measures if a sufficient number of particles is used. A number of 100 particles appears sufficient to produce good results while being computationally less expensive than using more particles.[†] The increased run-time of the tracking system when using pixel-cluster decomposition is mainly due to the additional cost of performing the data clustering. We note that the current implementation does not represent a computationally efficient implementation of the EM clustering method. Reducing the number of iterations in the EM algorithm decreases the run-time but degrades performance. For example, using only 5 EM iterations with 400 particles decreases run-time to 63.3 s but produces an average RMS position error per target of 31.8 m and RMS velocity error per target of 6.85 m/s.

	RMS(Pos)/Target [m]	RMS(Vel)/Target [m/s]	Run-time [s]
Regular Tracking	12,100	222.3	6
Multi-assignment	218.5	6.5	9.3
EM decomposition, 50 particles	37.7	7.1	23.3
EM decomposition, 100 particles	35.8	6.6	33.8
EM decomposition, 200 particles	27.7	6.2	55.5
EM decomposition, 400 particles	25.9	6.0	98.9

Table 1. Position and velocity estimation accuracy and run-time for regular tracking, pixel-cluster tracking using multi-assignment and pixel-cluster tracking using pixel-cluster tracking with the EM clustering algorithm and different numbers of particles.

Figure 11 compares the track state estimates projected into the focal plane of the unresolving sensor at $t = 150$ s for pixel-cluster tracking using (a) multi-assignment and (b) EM clustering (100 particles) based pixel-cluster decomposition. We can see that the EM based clustering pixel-cluster decomposition produces better track state estimates maintained on the subcomponents of the pixel-cluster.

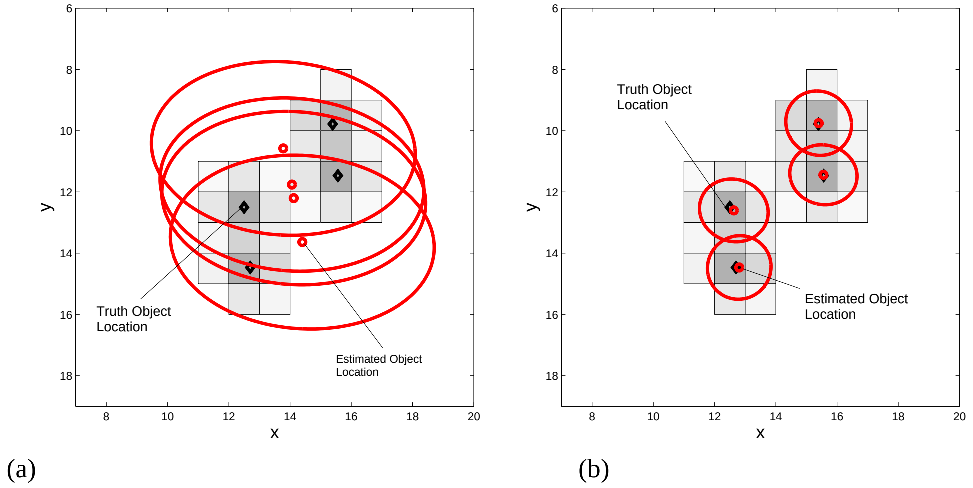


Figure 11. Track state estimates projected into the focal plane of the unresolving sensor at $t = 150$ s for pixel-cluster tracking using (a) multi-assignment and (b) EM clustering based pixel-cluster decomposition.

[†] The number of particles may also be selected depending on n_{obj} and number of pixels in a pixel-cluster.

4.4.3. Preventing Track Corruption

In Section 3.2.2 we illustrated that tracking performance can be degraded if an UCSO measurements updated a track that is also updated regularly by resolved object measurements. We refer to such measurements as corrupting measurements (see Section 3.2.2). In this paragraph we illustrate how pixel-cluster tracking with pixel-cluster parsing can be used to improve tracking performance in such scenarios. All results are based on 10 Monte Carlo runs. We consider a scenario similar to the scenario simulated for Study I in Section 4.4.2 above but with a second resolving sensor added to the scenario. Table 2 shows the RMS position and velocity estimation accuracy on the two tracks for the scenario with only the two resolving sensors and with all three sensors active. We can see that regular tracking with all three sensors active produces position estimation accuracy worse than when only the two resolving sensors are active.

	Track 1		Track 2	
	RMS(Pos) [m]	RMS(Vel) [m/s]	RMS(Pos) [m]	RMS(Vel) [m/s]
Two Resolving Sensors Only				
Regular Tracking	38.5	2.5	38.5	2.0
All Three Sensors				
Regular Tracking	46.4	2.2	41	2.1
Multi-Assignment	48.2	1.9	51.7	1.5
Pixel-Cluster Decomposition	15.8	1.6	21.1	1.4

Table 2. RMS position and velocity estimation accuracy on the two tracks for the scenario with only the two resolving sensors and with all three sensors active.

It has been suggested that multi-assignment of measurement-to-tracks is the desired approach when tracking in the presence of UCSOs.[17] However, in the presence of corrupting measurements, the multi-assignment of such a measurement to the tracks it associates with can actually degrade the tracking performance further. We can see in Table 2 that using multi-assignment, the position accuracy on both tracks degrades compared to regular tracking. The multi-assignment slightly improves only the velocity estimates. A better approach is to decompose the pixel-cluster corresponding to the potentially corrupting measurements into subclusters consistent with the tracks the measurements associates with. This can be seen in Table 2 where we see that pixel-cluster tracking with pixel-cluster decomposition produces track accuracy much improved compared to tracking only with the resolving sensors or when performing multi-assignment.

4.5. Application to Super-resolution

The pixel-cluster tracking methods as discussed above can produce accurate tracks throughout the system of sensors consistent with the most resolving sensor in the scenario. However, during most of midcourse it can be expected that even the best sensor will not be able to fully resolve all objects. Also, considering that high-resolution sensors are expensive and that current sensor design has almost reached the point were pixel size can not be decreased further due to shot noise limitation, it is important to consider new algorithms that can improve resolution beyond the physical resolution afforded by the sensor system, i.e., that can super-resolve CSOs. One can distinguish (1) single-image super-resolution techniques, and (2) multi-image super-resolution techniques.

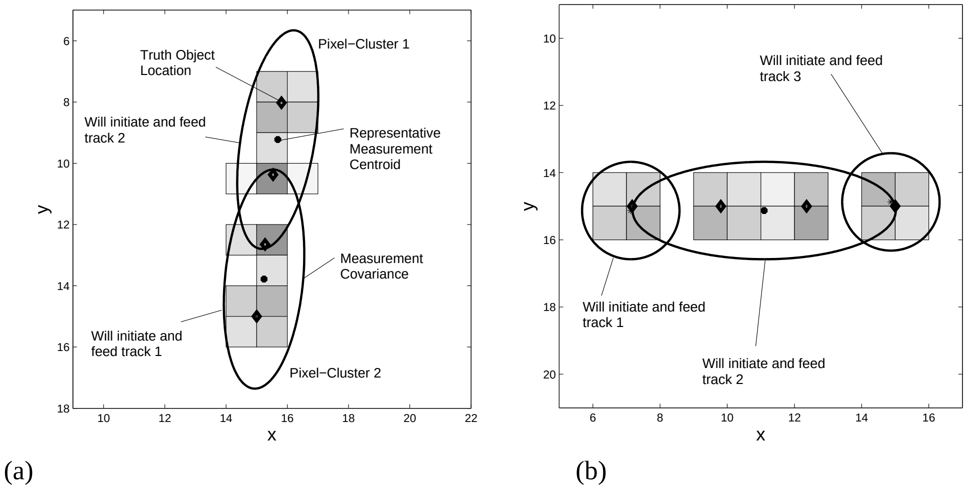
The single-image super-resolution techniques attempt to recover the original point sources by restoring the image.[1] These methods include de-blurring and blind-deconvolution methods. Reagan and Abatzoglou[5] suggested a model-based single-image super-resolution technique. A Bayesian technique to resolve closely spaced objects has been developed by Lillo and Schulenberg.[6] A similar approach can be obtained with the above pixel-cluster decomposition algorithm by treating the number of objects within a pixel-cluster as hypotheses and using an hypothesis testing approach that compares the value of different hypotheses, e.g., by using a Bayesian Information Criterion (BIC)[21] to assess the value of a decomposition.

Multi-image super-resolution techniques[14] attempt to combine multiple low resolution images to form a higher resolution image. This is related to methods of image fusion.[15] These multi-image super-resolution techniques are interesting since they may improve sensor resolution beyond the physical limitations of the sensor. The pixel-cluster tracking methods discussed above may support such advanced signal processing techniques since they provide an accurate association of pixel-clusters between frames and sensors. Thus, pixel-cluster tracking in support of super-resolution signal processing is a topic that may be investigated in future work. We also note that the multiple hypothesis decomposition approach discussed in the following section may offer super-resolution capability.

5. MULTIPLE HYPOTHESIS MULTIPLE FRAME PIXEL-CLUSTER DECOMPOSITION

The pixel-cluster tracking decomposition method as discussed in the previous section can produce accurate tracks throughout the system of sensors consistent with the most resolving sensor in the scenario. In this section we generalize the approach to consider multiple pixel-cluster decomposition per frame. We also formulate the problem in the framework of an explicit enumerative assignment problem over multiple frames of data that allows us to determine the optimal pixel-cluster decomposition from the information of multiple frames of data.

The pixel-cluster tracking method discussed in the previous section relied on using only the “best” multi-assignment for a frame of data and a set of established tracks. However, typically, the multi-assignment in the presence of closely spaced objects will produce several solutions that are almost equally likely. To illustrate this we consider a scenario that simulates four CSOs viewed by two sensors. Sensor 1 produces two UCSO pixel-clusters while Sensor 2 produces two resolved object returns and one UCSO pixel-cluster measurement. Figure 12 shows the focal planes of the two sensors at an early time in the scenario.


Pixel−Cluster 1
Truth Object Location
Will initiate and feed track 2
Representative Measurement Centroid
Measurement Covariance
Will initiate and feed track 1
Pixel−Cluster 2
y
x
6
8
10
12
14
16
18
10
12
14
16
18
20
22
Will initiate and feed track 3
Will initiate and feed track 1
Will initiate and feed track 2
y
x
10
12
14
16
18
20
6
8
10
12
14
16

(a) (b)

Figure 12. Scenario to illustrate the multiple hypothesis multiple frame pixel-cluster decomposition approach. Sensor 1 produces two UCSO pixel-clusters while Sensor 2 produces two resolved object returns and one UCSO pixel-cluster measurement.

Figure 13(a) shows the (lowest cost) multi-assignment obtained in this example for a multi-assignment between the three established tracks and the two pixel-clusters observed on Sensor 1 at the first time in the scenario where a multi-assignment was produced as a feasible solution. In this example, both the multi-view track 1 and the mono-view track 3 would assign with pixel-cluster 2. Thus, we would say that the observed measurement from pixel-cluster 2 is a candidate UCSO measurement with $n_{\text{obj}} = 2$ and decompose this pixel-cluster into two subclusters.

However, considering only the decomposition based on the “best” multi-assignment may not produce the best results when considering multiple frames of data. In the example shown in Figure 12 it is likely that the multi-assignment shown in Figure 13(b) produces a score similar to the score of the assignment shown in Figure 13(a). Also we may include a solution as shown in Figure 13(c). Thus, an *improved pixel-cluster decomposition approach may be formulated by considering multiple pixel-cluster decomposition hypotheses per frame*. The best decomposition may then be selected over multiple frames of data. As an example consider the four frames of data and pixel-cluster decomposition hypotheses

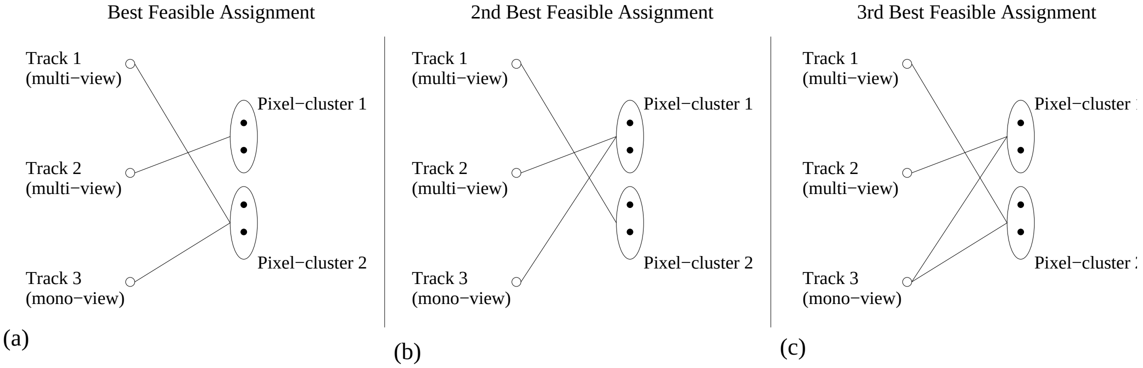


Figure 13. Multi-assignments of three established tracks with the two pixel-clusters observed on Sensor 1. (a) Possible best assignment, (b) possible assignment with 2nd best score, and (c) assignment possibly ranked third.

shown in Figure 14. The solution of this assignment problem can be achieved by solving 24 four dimensional one-to-one assignment problems.

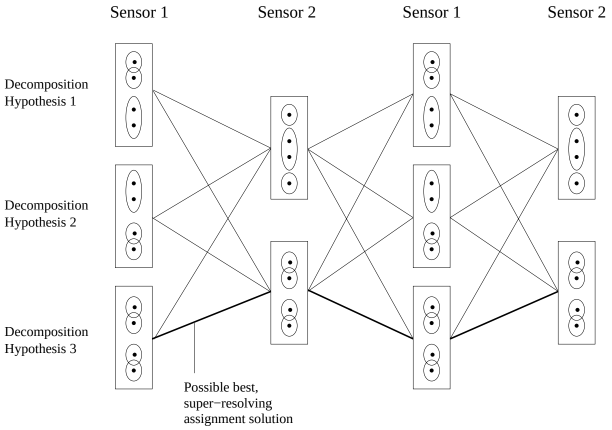


Figure 14: Four frames of data from two sensors and multiple pixel-cluster decomposition hypotheses per frame.

The following gives an mathematical formulation of this pixel-cluster decomposition assignment problem to match decomposition hypotheses between frames. We assume that we start with two lists of objects (e.g., candidate UCSO pixel-cluster of two frames of data). In a first step we compute a set of candidate pixel-cluster decompositions of the two data lists using a k-best multi-assignment approach to provide the necessary information to produce alternative decomposition hypotheses.

DEFINITION 5.1. *Let K and L denote two lists of objects and let $\mathcal{H}(K) = \{H_i(K)\}_{i\in I}$ and $\mathcal{H}(L) = \{H_j(L)\}_{j\in J}$ denote collections of pixel-cluster decompositions of K and L, respectively.*

One possible formulation of the pixel-cluster decomposition assignment problem is to determine the best score amongst the different decompositions in $\mathcal{H}(K)$ and a decomposition in $\mathcal{H}(L)$. Let $H_i(K) = \{K_{ik}\}_{k=1}^{P_i}$ and $H_j(L) = \{L_{jl}\}_{l=1}^{Q_j}$, e.g., $H_i(K) = \{K_{ik}\}_{k=1}^{P_i}$ denotes a decomposition of the data frame K into P_i subclusters (with $P_i \geq |K|$). Further we denote the pixel-clusters to that a pixel-cluster $i \in H_i(K)$ can be assigned to in $H_j(L)$ by the set $A(i)$, and the pixel-clusters in $H_i(K)$ to that a $j \in H_j(L)$ can be be assigned to by the set $B(j)$ (the computation of these sets allows us to keep only feasible associations and is accomplished through the use of gating methods). If we allow the computed pixel-cluster subclusters K_{ik} and L_{jl} to be assigned to only one subclusters on a different frame, then the assignment problem between $H_i(K)$ and $H_j(L)$ is

$$\begin{aligned}
\text{Minimize} \quad & \sum_{(k,l)\in\mathcal{A}} c_{kl}^{ij} x_{kl}, \\
\text{Subject To:} \quad & \sum_{l\in A(k)} x_{kl} \leq 1 \ \ (k = 1, ..., P_i), \\
& \sum_{k\in B(l)} x_{kl} \leq 1 \ \ (l = 1, ..., Q_j), \\
& x_{ij} \in \{0, 1\}.
\end{aligned}$$

Having computed the optimal score for each pairing (i, j), one chooses the one with the best score from this list. Note that if the number of complete clusterings in $\mathcal{H}(K)$ is M and the number in $\mathcal{H}(L)$ is N, then one must solve $M \times N$ assignment problems, i.e., this is an explicit enumeration scheme; however, if M and N are small, the method is very fast and suitable. The cost coefficients c_{kl}^{ij} between clustering hypotheses are computed in the tracking system through a likelihood function similar as the cost coefficients are computed for individual object tracking[22] by replacing individual object measurements and covariances with pixel-cluster representative centroids and pixel-cluster covariances.

We note that the above formulation is equivalent to the group-cluster tracking assignment problem to match group-cluster hypotheses between frames for multiple frame cluster tracking that was illustrated in previous work.[7,8] This formulation of the assignment problem can be generalized to the matching of pixel-clusters between frames that does not require an explicit enumeration as was previously shown for the group-cluster tracking problem.[7] This more general pixel-cluster decomposition assignment problem will be presented in future work.

6. CONCLUSION

In this paper we demonstrated an approach to pixel-cluster decomposition of unresolved objects in the focal plance of a visible/IR sensor that can greatly improve tracking in the presence of UCSOs. Compared to multi-assignment, pixel-cluster tracking with pixel-cluster decomposition produces more accurate tracks on all tracked subclusters throughout the constellation of sensors. We illustrated an extension of the approach to a multiple hypothesis pixel-cluster decomposition approach that fits the decomposition problem within a multiple frame group-cluster tracking framework developed in previous work.

Using pixel-cluster tracking, track state estimates projected onto the focal planes of a sensor allow us to accurately associate the observed measurements from other sensors with the observed measurement on this sensor. This association gives the potential to perform pixel-cluster image fusion and to aid super-resolution algorithms. The pixel-cluster decomposition algorithm presented can be extended to this purpose when information about the sensor's point spread function is available. This aspect of the decomposition problem that aims at resolution into resolved objects beyond the resolution afforded by the sensors will be presented in future work.

ACKNOWLEDGMENTS

We wish to thank the modelling and simulation team at Spectrum Astro for their continuing support and assistance. Specifically we wish to thank Spectrum Astro for providing us with a simulator prototype for simulating image processing of infrared data. This work was also supported in part by the Missile Defense Agency through Contract Number DASG60-03-P-0042.

REFERENCES

1. M. Banham and A. Katsaggelos, "Digital image restoration," *IEEE Signal Processing Magazine* , pp. 24–41, March 1997.
2. D. Sheppard, B. Hunt, and M. Marcellin, "Iterative multiframe super-resolution algorithms for atmospheric turbulencedegraded imagery," 1998.

3. S. Park, M. Park, and M. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine* **20**, pp. 21–36, 2003.
4. G. Bretthorst, "Bayesian analysis of signals from closely-spaced objects," in *Infrared Systems and Components III, Robert L. Casell ed., SPIE Vol. 1050*, pp. 93–104, 1989.
5. J. Reagan and T. Abatzoglou, "Model-based superresolution CSO processing," *Proceedings of the SPIE, Signal and Data Processing of Small Targets* **1954**, pp. 204–218, 1993.
6. W. Lillo and N. Schulenburg, "A Bayesian closely spaced object resolution technique," in *Proc. SPIE Vol. 2235*, pp. 2–13, 1994.
7. S. Gadaleta, M. Klusman, A. B. Poore, and B. J. Slocumb, "Multiple frame cluster tracking," in *SPIE Vol. 4728, Signal and Data Processing of Small Targets*, pp. 275–289, 2002.
8. S. Gadaleta, A. B. Poore, S. Roberts, and B. J. Slocumb, "Multiple Hypothesis Clustering Multiple Frame Assignment tracking," in *SPIE Vol. 5204, Signal and Data Processing of Small Targets*, August 2003.
9. X. Liu, "CMOS image sensors dynamic range and snr enhancement via statistical signal processing," tech. rep., Stanford University, 2002. Ph.D Thesis.
10. R. Gonzalez and R. Woods, *Digital Image Processing*, Pearson Education, 2002.
11. T. Veldhuizen, "Grid filters for local nonlinear image restoration," tech. rep., Dept. of Systems Design Engineering, University of Waterloo, 1998. Master's Thesis.
12. Y. Yardimci, J. Cadzow, and M. Zhu, "Comparative study of high resolution algorithms for multiple point source location via infrared focal plane arrays," *Proceedings of the SPIE, Signal and Data Processing of Small Targets* **1954**, pp. 59–69, 1993.
13. W. Benenson, J. Harris, H. Stocker, and H. Lutz, *Handbook of Physics*, Springer-Verlag, New York, 2002.
14. M. Kang and S. Chaudhuri, "Super-resolution image reconstruction," *IEEE Signal Processing Magazine* **20**, pp. 19–20, 2003.
15. H. Wang, J. Peng, and W. Wu, "Fusion algorithm for multisensor images based on discrete multiwavelet transform," in *IEE Proc.-Vis. Image Signal Process., Vol. 149*, pp. 283–289, 2002.
16. B. Masson and C. Pfeiffer, "The MSX on-board signal and data processor (OSDP)," in *Proceedings SPIE Vol. 1698, Signal and Data Processing of Small Targets*, pp. 211–221, 1992.
17. T. Kirubarajan, Y. Bar-Shalom, and K. Pattipati, "Multiassignment for tracking a large number of overlapping objects," *IEEE Transactions on Aerospace and Electronic Systems* **37**, pp. 2–21, 2001.
18. W. Richardson, "Bayesian-based iterative method of image restoration," *Journal of the Optical Society of America* , pp. 55–59, 1972.
19. L. Lucy, "An iterative technique for the rectification of observed distribution," *The Astronomical Journal* , pp. 745–754, 1974.
20. A. Dempster, N. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, B* **39**, p. 39, 1977.
21. G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, Wiley, New York, 1997.
22. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, MA, 1999.